Project 1's Software Architecture

Team 4's Evaluation

Upon analysis, the *Battleship* project we inherited seems to be following the **N-Tier** Architecture. Their codebase is divided up to serve two primary functions, ***logic code*** and ***visual representation***. It would appear that initially they wrote their code in Python but weren't able to get it to function properly so switched to a C++ terminal interface. However, even with this change in language, the organization of the code and function didn't change (albeit unfinished, either due to being unable or not having enough time to finish the development).

Why **N-Tier** and not **3-Tier**? Well as explained above, the code is divided up between logic and visual representation. A **3-Tier** Architecture requires a *Presentation tier* (visual representation), *Logic tier* (logic code). And *Data tier* (storage of data). Unfortunately, this project doesn't exactly *store* data. True, the program needs to keep track of array positioning, variables, and loop status (all of which could be argued as a means of storage), but in this case data storage refers to long-term storage, or storage of larger sizes. Typically, examples of this refer to client information, program preferences, essentially something that remains in the program's data storage after the program has been restarted. This of course isn't the case with this project. Assuming the project we inherited worked by default, there isn't a function that keeps track of wins/losses over multiple program iterations or an AI opponent that uses machine-learning to play better against an opponent.

Is **N-Tier** the best option for this project? For the scale and time of development for this project, I believe it is. I believe that most groups will have adopted this model of software architecture as well, as it proves to be the most sustainable for the list of given requirements. Sure, **3-Tier** could improve upon the project for the reasons listed above, but that would overly elongate the development cycle and complexity of the project.